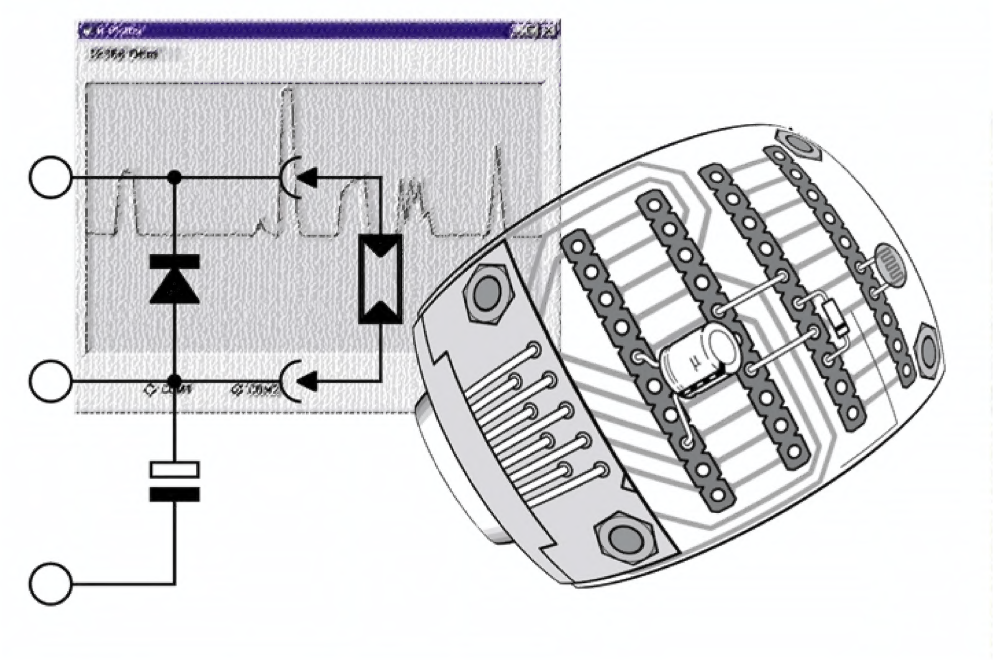
PC Serial Peripheral Design (6)

Measurements with sensors



By B. Kainka

In the previous instalment we measured capacitances and resistances. You may have wondered if this was all worth the effort, when accurate and inexpensive digital multimeters are readily available, but we shall see that we are able to take readings from a wide variety of sensors by measuring their resistance.

The best demonstration involves a light-dependent resistor (LDR). **Figure 1** shows a simple light-measuring circuit; **Figure 2** shows how it might be constructed. In this case we are frequently interested in how the reading changes over time, and this is best shown on a graph.

Light measurement

In order to produce a graph on the screen using Visual Basic, we need to use a so-called PictureBox, which can be dragged from the form's toolbar. It is very important to set the correct size. Visual Basic can use a variety of units and here we want to measure the size in pixels. The ScaleMode property should be set to '3 - pixel', and the box can be

Listing I. Resistance Plotter

```
Dim y1, y2, x1, x2, n
Private Sub Form_Load()
 i = OPENCOM(“COM2,1200,N,8,1”)
 If i = 0 Then
    i = OPENCOM(“COM1,1200,N,8,1”)
    Option1.Value = True
 End If
 If i = 0 Then MsgBox (“COM Interface Error”)
 TXD 0
 RTS 0
 DTR 0
 Counter1 = 0
 Timer1.Interval = 1000
 n = 0
End Sub

Private Sub Form_Unload(Cancel As Integer)
  CLOSECOM
End Sub

Private Sub Option1_Click()
 i = OPENCOM(“COM1,1200,N,8,1”)
 If i = 0 Then MsgBox (“COM1 not available”)
End Sub

Private Sub Option2_Click()
 i = OPENCOM(“COM2,1200,N,8,1”)
 If i = 0 Then MsgBox (“COM2 not available”)
End Sub

Private Sub Timer1_Timer()
  DTR 1
  REALTIME (True)
  TIMEINITUS
  While (DSR() = 0) And (TIMEREADUS() < 900000)
  Wend
  T = TIMEREADUS()
  DTR 0
  T = T * 0.932
  R = 2200 + 7800 * (T - 76300) / (294600 - 76300)
  REALTIME (False)
  R = Int(R)
  Label1.Caption = Str$(R) + “ Ohm”
  y2 = 300 - R / 100
  If n = 0 Then y1 = y2
  x1 = n
  n = n + 5
  x2 = n
  Picture1.Line (x1, y1)-(x2, y2)
  y1 = y2
End Sub
```

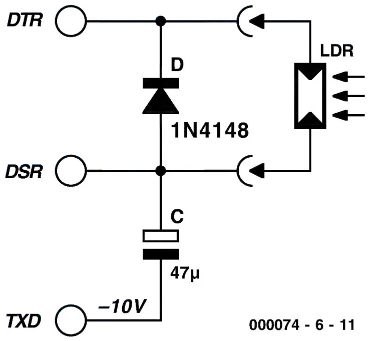


Figure 1. Measurement using a light-dependent resistor

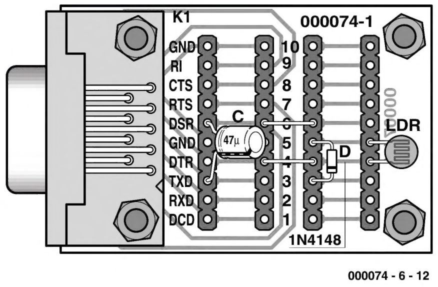


Figure 2. Construction of the LDR circuit

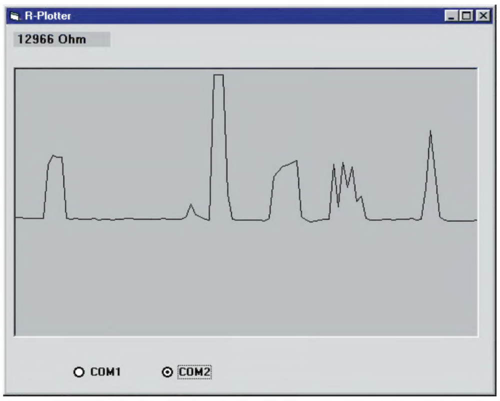


Figure 3. Results from the LDR measurement

dragged out to the correct size using the mouse. For the present example (plotter1.frm) a PictureBox 300 by 500 points is required.

We can draw in the PictureBox using the Line command: see **Listing 1** for an example. The plot is drawn over a period of 100 seconds, the x-coordinate advancing by 5 pixels as each reading is taken. A global variable n must be specially declared to store the current position across calls to the timer procedure. On the y-axis resistance (R/100) is plotted, with a maximum value of 300 (30000 Ω=30 kΩ). Every second a new value is plotted in the graph in **Figure 3** and a new line segment is drawn from the last point to the new one. The old coordinate values must be stored, and global variables are used for this purpose.

Skin resistance measurements

Another interesting ‘sensor’ is the human skin: a pair of bare wires can be wound around two fingers. The resistance of the skin is not constant,

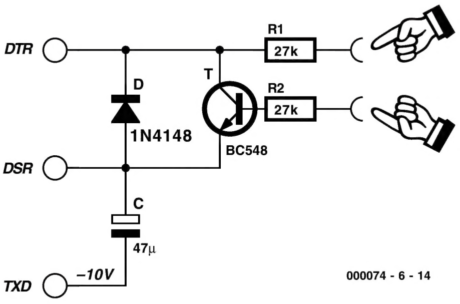


Figure 4. Amplifier for measuring skin resistance

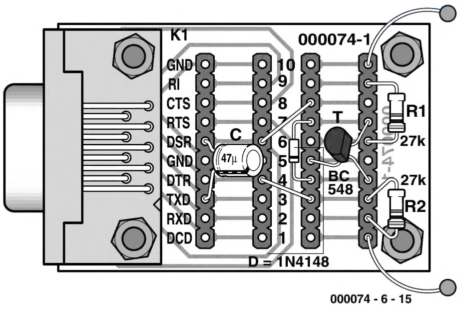


Figure 5. Measuring skin resistance

but varies according to its moisture levels. When people lie, they sweat: so we can build a lie detector by determining when the skin resistance falls. The problem is that skin resistance is typically rather higher than the maximum value we can measure with our circuit: we need to use a transistor for amplification (**Figure 4, Figure 5**).

An NPN transistor amplifies the small current flowing through the skin, allowing our simple A/D converter to measure relatively high resistances. The input to the circuit is protected by two additional resistors, so that the measured resistance will not be too small even if the wires are shorted. The reading obtained depends on the contact area of the wires and the moisture level of the skin. The results can be viewed using the resistance plotter program.

Listing 2. Extensions for temperature measurement

```
Private Sub Timer1_Timer()
  DTR 1
  REALTIME (True)
  TIMEINITUS
  While (DSR() = 0) And (TIMEREADUS() < 1500000)
  Wend
  T = TIMEREADUS()
  T = T * 1.0000000001
  R = 2200 + 7800 * (T - 76300) / (294600 - 76300)
  REALTIME (False)
  R = Int(R)
  Temp = 1 / (Log(R / 10000) / 4300 + 1 / 298) - 273
  Temp = Int(Temp * 10) / 10
  Label1.Caption = Str$(Temp) + “°C”
  DTR 0
End Sub
```

Listing 3. Conversion to the Fahrenheit scale

Although the German physicist Gabriel Daniel Fahrenheit died in 1736, some people are still more comfortable working with temperatures measured in Fahrenheit. They need only change a couple of lines of the program, as follows:

```
  Temp = 32 + (Temp / 100 * 180)
  Temp = Int(Temp * 10) / 10
  Label1.Caption = Str$(Temp) + “ F”
```

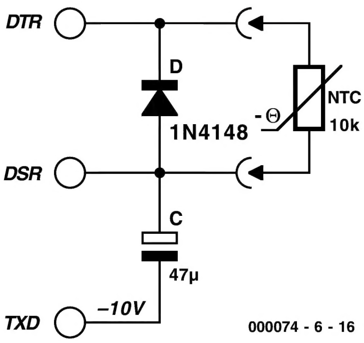
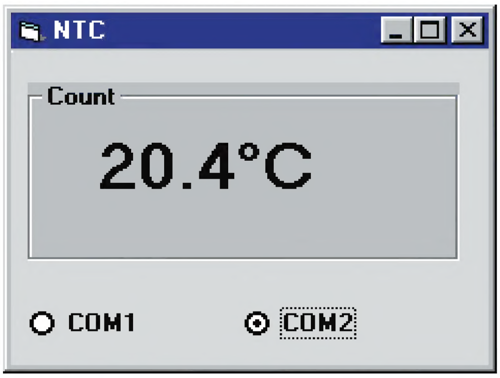


Figure 6. Connecting an NTC thermistor



Count

20.4°C

○ COM1 ● COM2

Figure 7. Temperature display

Temperature measurement

Temperature can also be measured using our resistance measuring circuit. The circuit of **Figure 6** shows how easy it is to connect a 10 kΩ NTC thermistor to build a very simple yet perfectly usable thermometer.

The resistance characteristic of an NTC thermistor can be approximated by an exponential curve. In the following formula, T is the absolute temperature in Kelvin (T/°C + 273), while B is a value, typically between 2000 and 500 Kelvin, provided by the thermistor manufacturer. In fact the value of B is not perfectly constant but rises gradually with temperature: for this reason the value $B_{25/85}$ is often quoted, where calibration has been performed at 25°C and 85°C.

$$R(T) = R_{25} \cdot e^{B\cdot\left(\frac{1}{T}-\frac{1}{298K}\right)}$$

An NTC thermistor is therefore specified to a first approximation by two values: B and the thermistor’s nominal resistance R_{25}. For a typical 10 kΩ NTC thermistor the value of B might be 4300 K. Rearranging the expression above and substituting R_{25} = 10 kΩ we can derive the following line of Visual Basic to calculate the value of T in Celsius:

Temp = 1 / (Log(R / 10000) / 4300 + 1 / 298) – 273

The VB ‘Log’ function calculates the natural logarithm, often abbreviated to ‘ln’ in mathematics textbooks.

In the program shown in **Listing 2** we first carry out a resistance measurement and then convert the measured value into temperature. The results appear to one decimal place as shown in **Figure 7**. Finally, **Listing 3** shows how to convert the reading from the Celsius scale into Fahrenheit.

(000074-6)